This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (Currently amended): A method for generating a dump file, the method comprising:

a. generating a minidump file that does not include all ~~operating system data~~ volatile system memory containing ~~by gathering~~ at least:

i. thread information for at least one running thread,

ii. context information for the thread,

iii. callstack information for the thread,

iv. process information for a process in which the thread is running, and

v. information identifying a reason comprising one of the following reasons: callstack fault, processor fault, and application program fault, for generating the minidump file; and

b. storing the minidump file to a storage medium.

Claim 2 (Previously presented): The method as recited in Claim 1, further comprising determining when to generate the minidump file.

Claim 3 (Previously presented): The method as recited in Claim 1, wherein generating the minidump file further includes gathering processor information about at least one processor.

Claim 4 (Previously presented): The method as recited in Claim 2, wherein determining when to generate the minidump file further includes determining that an exception has occurred.

Claim 5 (Previously presented): The method as recited in Claim 1, wherein the minidump file does not include data stored in global initialized memory.

Claim 6 (Previously presented): The method as recited in Claim 1, wherein the minidump file does not include data stored in uninitialized memory.

Claim 7 (Previously presented): The method as recited in Claim 1, wherein the minidump file does not include executable instructions used by a processor to execute a program.

Claim 8 (Previously presented): The method as recited in Claim 1, wherein the minidump file is a kernel minidump file associated with an operating system and the at least one running thread is the single thread which encountered an exception.

Claim 9 (Previously presented): The method as recited in Claim 8, wherein the callstack information includes kernel stack information.

Claim 10 (Previously presented): The method as recited in Claim 1, wherein the process information identifies a process that initiated the thread.

Claim 11 (Previously presented): The method as recited in Claim 1, further comprising:

allocating a buffer space in memory during an initialization process, wherein the buffer space is suitable for storing the gathered information; and

reserving space on the storage medium suitable for writing the contents of the buffer space.

Claim 12 (Previously presented): The method as recited in Claim 11, wherein generating the minidump file further includes initially storing the thread information, the context information, the callstack information, the process information, and the information identifying the reason for generating the minidump file to the buffer space, and then copying the minidump file from the buffer space to the storage medium.

Claim 13 (Previously presented): The method as recited in Claim 12, further comprising upon re-initialization, after having stored the minidump file to the storage medium, accessing the minidump file on the storage medium and using at least a portion of the minidump file to further understand an exception that was at least one reason for generating the minidump file.

Claim 14 (Previously presented): The method as recited in Claim 1, wherein the minidump file is a user minidump file associated with at least one non-operating system program.

Claim 15 (Previously presented): The method as recited in Claim 1, wherein generating the minidump file further includes gathering callstack information for all running threads.

Claim 16 (Previously presented): The method as recited in Claim 15, wherein the callstack information includes a user callstack.

Claim 17 (Previously presented): The method as recited in Claim 1, wherein generating the minidump file further includes gathering processor context information for all running threads.

Claim 18 (Previously presented): The method as recited in Claim 1, wherein generating the minidump file further includes gathering a listing of loaded modules for a faulting application program.

Claim 19 (Previously presented): The method as recited in Claim 1, wherein the minidump file is a directory indexed file that uses relative virtual addresses (RVAs).

Claim 20 (Currently amended): A computer-readable medium having computer-executable instructions for causing at least one processor to perform acts comprising:

gathering minidump file information that does not include all ~~operating system data~~ volatile system memory but does include at least thread information for at least one running thread, context information for the thread, callstack information for the thread, process information for the process in which the thread is running, and information identifying a reason comprising one of the following reasons: callstack fault, processor fault, and application program fault, for generating the minidump file.

Claim 21 (Previously presented): The computer-readable medium as recited in Claim 20, wherein generating the minidump file further includes storing the dump file to a storage medium.

Claim 22 (Previously presented): The computer-readable medium as recited in Claim 20, wherein gathering the minidump file information further includes gathering processor information about at least one processor.

Claim 23 (Previously presented): The computer-readable medium as recited in Claim 20, having further computer-executable instructions for causing the at least one processor to perform acts comprising determining when to generate the minidump file.

Claim 24 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the minidump file does not include data stored in global initialized memory.

Claim 25 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the minidump file does not include data stored in uninitialized memory.

Claim 26 (Previously presented): The computer-readable medium as recited Claim 24 wherein the minidump file does not include executable instructions used by the at least one processor to execute a program.

Claim 27 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the minidump file is a kernel minidump file associated with an operating system and the at least one running thread is the single thread which encountered an exception.

Claim 28 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the callstack information includes kernel stack information.

Claim 29 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the process information identifies a process that initiated the thread.

Claim 30 (Previously presented): The computer-readable medium as recited in Claim 20, further comprising computer-executable instructions for causing the at least one processor to perform acts comprising:

allocating a buffer space in memory during an initialization process, wherein the buffer space is suitable for storing the minidump file information; and

reserving space on a storage medium drive suitable for writing the contents of the buffer space.

Claim 31 (Previously presented): The computer-readable medium as recited in Claim 30, wherein generating the minidump file further includes initially storing the thread information, the context information, the callstack information, the process information, and the information identifying the reason for generating the minidump file to the buffer space, and then copying the minidump file from the buffer space to the storage medium.

Claim 32 (Previously presented): The computer-readable medium as recited in Claim 31, further comprising computer-executable instructions for causing the at least one processor to perform acts comprising, upon re-initialization after having stored the minidump file to the storage medium, accessing the minidump file on the storage medium and using at least a portion of the minidump file to further understand an exception that was at least one reason for generating the minidump file.

Claim 33 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the minidump file is a user minidump file associated with at least one non-operating system program.

Claim 34 (Previously presented): The computer-readable medium as recited in Claim 20, wherein gathering the minidump file information further includes gathering callstack information for all running threads.

Claim 35 (Previously presented): The computer-readable medium as recited in Claim 34, wherein the callstack information includes a user callstack.

Claim 36 (Previously presented): The computer-readable medium as recited in Claim 20, wherein gathering the minidump file information further includes gathering processor context information for all running threads.

Claim 37 (Previously presented): The computer-readable medium as recited in Claim 20, wherein gathering the minidump file information further includes gathering a listing of all loaded modules for the faulting application program.

Claim 38 (Previously presented): The computer-readable medium as recited in Claim 20, wherein the minidump file is a directory indexed file that uses relative virtual addresses (RVAs).

Claim 39 (Currently amended): An apparatus comprising:

memory;

a data storage drive configured to write data files to at least one data storage medium; and

at least one processor operatively coupled to the memory and the data storage drive and configured to:

a. generate a minidump file that does not include all ~~operating system data~~ volatile system memory containing ~~by gathering in the memory~~ at least:

i. thread information for at least one running thread,

ii. context information for the thread,

iii. callstack information for the thread,

iv. process information for the process in which the thread is running, and

v. information identifying a reason comprising one of the following reasons: callstack fault, processor fault, and application program fault, for generating the minidump file, and

b. store the minidump file to the storage medium.

Claim 40 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to determine when to generate the minidump file.

Claim 41 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to gather processor information about the at least one processor and include the processor information in the minidump file.

Claim 42 (Previously presented): The apparatus as recited in Claim 40, wherein the at least one processor is further configured to determining when to generate the minidump file based on an exception.

Claim 43 (Previously presented): The apparatus as recited in Claim 39, wherein the minidump file does not include data stored in global initialized memory.

Claim 44 (Previously presented): The apparatus as recited in Claim 39, wherein the minidump file does not include data stored in uninitialized memory.

Claim 45 (Previously presented): The apparatus as recited Claim 39 wherein the minidump file does not include executable instructions used by the at least one processor to execute a program.

Claim 46 (Previously presented): The apparatus as recited in Claim 39, wherein the minidump file is a kernel minidump file associated with an operating system and the at least one running thread is the single thread which encountered an exception.

Claim 47 (Previously presented): The apparatus as recited in Claim 39, wherein the callstack information includes kernel stack information.

Claim 48 (Previously presented): The apparatus as recited in Claim 39, wherein the process information identifies a process that initiated the thread.

Claim 49 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to:

allocate a buffer space in the memory during an initialization process; and

reserve space on the storage medium drive suitable for writing the contents of the buffer space.

Claim 50 (Previously presented): The apparatus as recited in Claim 49, wherein the at least one processor is further configured to:

generate the minidump file by initially storing the thread information, the context information, the callstack information, the process information, and the information identifying the reason for generating the dump file to the buffer space, and then copying the minidump file from the buffer space to the storage.

Claim 51 (Previously presented): The apparatus as recited in Claim 50, wherein the at least one processor is further configured to, upon re-initialization after having stored the minidump file to the storage medium, access the minidump file on the storage medium and use at least a portion of the minidump file to further understand an exception that was at least one reason for generating the minidump file.

Claim 52 (Previously presented): The apparatus as recited in Claim 39, wherein the minidump file is a user minidump file associated with at least one non-operating system program.

Claim 53 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to gather callstack information for all running threads as part of the minidump file.

Claim 54 (Previously presented): The apparatus as recited in Claim 53, wherein the callstack information includes a user callstack.

Claim 55 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is configured to gather processor context information for all running threads as part of the minidump file.

Claim 56 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is configured to gather a listing of all loaded modules for a faulting application program as part of the minidump file.

Claim 57 (Previously presented): The apparatus as recited in Claim 39, wherein the minidump file is a directory indexed file that uses relative virtual addresses (RVAs).

Claims 58-66 (Canceled)

Claim 67 (Previously presented): The method as recited in Claim 1, further comprising providing the minidump file to at least one external device.

Claim 68 (Previously presented): The method as recited in Claim 12, upon system re-initialization, transferring the minidump file from the storage medium to at least one external device.

Claim 69 (Previously presented): The method as recited in Claim 1, wherein generating the minidump file further includes gathering a list of loaded modules.

Claim 70 (Previously presented): The computer-readable medium as recited in Claim 20, having further computer-executable instructions for causing the at least one processor to perform acts comprising providing the minidump file to at least one external device.

Claim 71 (Previously presented): The computer-readable medium as recited in Claim 30, having further computer-executable instructions for causing the at least one processor to perform acts comprising, upon system re-initialization, transferring the minidump file from the storage medium to at least one external device.

Claim 72 (Previously presented): The computer-readable medium as recited in Claim 20, wherein gathering the minidump file information further includes gathering a list of loaded modules.

Claim 73 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to provide the minidump file to at least one external device.

Claim 74 (Previously presented): The apparatus as recited in Claim 49, wherein the at least one processor is further configured to, upon system re-initialization, transferring the minidump file from the storage medium to at least one external device.

Claim 75 (Previously presented): The apparatus as recited in Claim 39, wherein the at least one processor is further configured to gather a list of loaded modules as part of the minidump file.

Claims 76-77 (Canceled)